UNIVERSIDADE ESTADUAL DE MARINGÁ
PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA – PIBIC/CNPq-Fundação Araucária-UEM
DEPARTAMENTO DE MATEMÁTICA
ORIENTADOR: Prof. Dr. Eduardo Brandani da Silva
Bolsista: Anderson Dario Arendt

COMPUTAÇÃO E INFORMAÇÃO QUÂNTICA – UMA INTRODUÇÃO

Maringá, 31 de julho de 2018

UNIVERSIDADE ESTADUAL DE MARINGÁ
PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA – PIBIC/CNPq-Fundação Araucária-UEM
DEPARTAMENTO DE MATEMÁTICA
ORIENTADOR: Prof. Dr. Eduardo Brandani da Silva
Bolsista: Anderson Dario Arendt

COMPUTAÇÃO E INFORMAÇÃO QUÂNTICA – UMA INTRODUÇÃO

Relatório contendo os resultados finais do projeto de iniciação científica vinculado ao PIBIC/CNPq-Fundação Araucária-UEM.

Maringá, 31 de julho de 2018

Resumo

O projeto teve como objetivo realizar o estudo sobre como a informação é representada e como computações são realizadas no contexto quântico utilizando-se da álgebra linear para teorizar tais conceitos. O foco foi em compreender a estruturação de circuitos lógicos e como sua combinação se dá para construir alguns algoritmos que são muito mais eficientes do que qualquer análogo algoritmo de computação clássica. Resultando assim em um estudo introdutório em aspectos de diversos contextos, tais como matemático, físico e computacional, a fim de gerar uma compreensão clara sobre a computação e informação quântica.

Palavras-chaves: Algoritmos Quânticos, Circuitos Quânticos, Qubits.

Sumário

1 INTRODUÇÃO

Muitas estratégias vêm sendo aplicadas para prover um maior desempenho nos computadores modernos, focadas principalmente em otimizar a arquitetura e organização dos processadores e memórias. Entretanto a técnica mais utilizada para conseguir maior desempenho – que é diminuindo o tamanho dos componentes (principalmente transistores), acarretando assim que a distância física que a corrente elétrica tenha que percorrer seja também menor – está próxima de se tornar impraticável. Isto ocorre porque esse processo de miniaturização de componentes está esbarrando em uma limitação física, com componentes alcançando tamanhos semelhantes aos de um átomo. Sendo assim, nessa escala, efeitos quânticos começam a influenciar na conformidade do desempenho dos componentes. Portanto, uma alternativa para contornar esse problema é desenvolver computadores quânticos, cujo princípio de operação se dá através da manipulação de elementos quânticos para representar a informação ao invés de utilizar corrente elétrica.

A ideia de construir um computador quântico envolve diversas questões complexas. Primeiramente é definir o que é bom e viável com características quânticas que possa representar a informação e a sua facilidade de sofrer manipulação. A segunda é como abstrair e manipular essa informação. Se o contexto é o mundo quântico, então uma forma de lidar com a abstração é usar das ferramentas já conhecidas da mecânica quântica. Sendo assim, conceitos já conhecidos da computação clássica podem ser adaptados para o da computação quântica, tais como bits, portas-lógicas, algoritmos e teoria de computação. Assim, conceitos de física e uma presença forte de álgebra linear se fazem necessários para lidar com tal tema.

Com a computação nesse novo paradigma, espera-se que ela seja capaz de resolver problemas que em computadores clássicos são muito custosos em termos de eficiência. De fato já há bons exemplos de algoritmos quânticos criados que reduzem consideravelmente a quantidade de passos necessários para solucionar um problema comparado com algoritmos clássicos, sendo exemplos os algoritmos de Shor e Deutsch (NIELSEN; CHUANG, 2011). Isso ocorre por causa do fenômeno conhecido como “superposição” no mundo quântico que, quando utilizado de forma inteligente, fazendo os análogos quânticos dos bits clássicos, (chamados qubits) entrarem nesse estado e aplicarem operações que aproveitem das características desse estado, acaba se tornando possível a realização de diversos cálculos simultaneamente (YANOFSKY; MANNUCCI, 2008).

Por conta das razões citadas acima, faz necessário que a ideia de construção de um computador e desenvolvimento de algoritmos sejam encaradas de uma forma completamente diferente do que se conhece classicamente, fazendo-se necessário o estudo e desenvolvimento cada vez mais técnicas para este meio que está recebendo cada vez mais atenção de empresas e órgãos de pesquisa de todo o mundo que vem nele o futuro da

computação.

2 OBJETIVOS

Os objetivos gerais são: introduzir o acadêmico no meio da produção científica a fim de uma formação acadêmica mais ampla. Já os objetivos específicos são: propiciar ao acadêmico a aptidão necessária para que avance qualitativamente nos conteúdos da grade curricular, desenvolvendo a capacidade de comunicação escrita e oral, propiciar a oportunidade de estudar conteúdos matemáticos dentro de um contexto interdisciplinar e incentivá-lo a prosseguir seus estudos em direção à pós-graduação.

3 DESENVOLVIMENTO

Foi realizado um levantamento bibliográfico sobre o tema, onde deu-se pela leitura de livros e artigos sobre os assuntos. Quanto a metodologia do projeto, foi escolhida uma abordagem típica da área de Matemática, consistindo de apresentações semanais de seminários com o orientador, visando discutir e aprofundar os assuntos estudados.

4 RESULTADOS

Nessa seção é apresentado os resultados do estudo realizado durante este projeto.

4.1 Preliminares

Para se trabalhar no contexto quântico de computação alguns conceitos devem ser introduzidos. Alguns deles são adaptações de conceitos já existentes no contexto clássico, por outro lado, alguns são novos dado as características exclusiva deste meio. Tais conceitos são introduzidos nas subseções abaixo.

4.1.1 Operadores Lineares

Operadores lineares são funções que agem sobre vetores. Dado um operador linear $A : V \to W$, A age linearmente sobre um vetor $|v\rangle$.

$$A|v\rangle = A\left(\sum_i a_i|v_i\rangle\right) = \sum_i a_i A|v_i\rangle \tag{1}$$

4.1.2 Produto Interno

O produto interno é uma função muito importante no contexto da álgebra linear como um todo. Essa função basicamente tem como entrada dois vetores e produz um

número – neste contexto, complexo – como saída. A notação utilizada para esse operador é $\langle v|w\rangle$, onde v e w são dois vetores qualquer. Sendo assim podemos definir o produto interno sobre um espaço vetorial $\mathbb{V}$ como:

$$\langle -|-\rangle : \mathbb{V} \times \mathbb{V} \longrightarrow \mathbb{C}$$

O produto interno respeita as seguintes propriedades:

- Simetria:
  $\langle v|w\rangle = \langle w|v\rangle^{\dagger}$
- Positividade:
  $\langle v|v\rangle > 0$
  $\langle v|v\rangle = 0$ se e somente se $|v\rangle$ for o vetor nulo do espaço vetorial.
- Adição:
  $\langle v + u|w\rangle = \langle v|w\rangle + \langle u|w\rangle$
  $\langle v|u + w\rangle = \langle v|u\rangle + \langle v|w\rangle$
- Multiplicação: linear no segundo argumento e conjugado linear no primeiro:

$$\langle v|\lambda w\rangle = \langle v|\sum_i \lambda_i w_i\rangle = \sum_i \lambda_i \langle v|w_i\rangle = \lambda\langle v|w\rangle$$

$$\langle \lambda v|w\rangle = \langle \sum_i \lambda_i^* v_i|w\rangle = \sum_i \lambda_i^* \langle v_i|w\rangle = \lambda^*\langle v|w\rangle$$

onde $\{|v\rangle, |w\rangle, |u\rangle\} \in \mathbb{V}$ e $\lambda \in \mathbb{C}$.

Com o produto vetorial é possível descobrir a componente de um $|v\rangle$ com respeito a uma determinada base $|v_i\rangle$. Seja $|v\rangle = a_1|v_1\rangle + ... + a_n|v_n\rangle$ então $\langle v_i|v\rangle = a_i$ para $1 \leq i \leq n$.

Além disso, para todo espaço vetorial $\mathbb{V}$ com produto interno definido é possível aplicar a função chamada *norma* (que pode ser interpretada como uma medida de tamanho sobre um vetor). Formalmente a norma é $|-| : \mathbb{V} \longrightarrow \mathbb{R}$ tal que:

$$||v\rangle| = \sqrt{\langle v|v\rangle}$$

respeitando as seguintes propriedades:

- Positividade:
  $||v\rangle| > 0$ se $|v\rangle \neq 0$ e $|0| = 0$.
- Desigualdade triangular:
  $||v\rangle + |w\rangle| \leq ||v\rangle| + ||w\rangle|$

- Multiplicação por escalar:
  $|\lambda|v\rangle| = \lambda||v\rangle|$

Quando os vetores da base de um espaço vetorial tiverem a norma unitária, isto é, igual a 1, chamamos esses vetores de *ortonomais* e a base como um todo de *base ortonormal*. Com essa condição é definido a *Função de Kronecker*. Essa função toma como entrada uma base ortonormal $\beta = \{|v_1\rangle, ..., |v_n\rangle\}$ e opera para cada par dois a dois deste conjunto. Assim temos:

$$\langle v_i|v_j\rangle = \delta_{ij} = \begin{cases} 1, & \text{se } i = j; \\ 0, & \text{se } i \neq j. \end{cases}$$

onde δ_{ij} denota a *função de Kronecker*.

4.1.3 Adjuntos e Hermitianos

Uma classe muito importante de operadores é conhecida como *operadores adjuntos* ou *operadores Hermitianos*. Um operador é A é dito hermitiano se $A = A^\dagger$. Logo, suponha que A opera em um espaço vetorial $\mathbb{V}$ complexo com produto interno. Assim, para todo $|v\rangle, |w\rangle \in \mathbb{V}$ temos:

$$\langle v|A|w\rangle = \langle v|A^\dagger|w\rangle$$

Se a composição de operadores é dita hermitiana então eles respeitam a relação $(AB)^\dagger = B^\dagger A^\dagger$. Por definição $\langle v| = |v\rangle^\dagger$. Então a aplicação $(A|v\rangle)^\dagger = |v\rangle^\dagger A^\dagger = \langle v|A^\dagger$.

4.1.4 Produto Tensorial

O *produto tensorial* é um operador que tem por objetivo unir dois espaços vetoriais e tem papel fundamental para construir operações entre sistemas quânticos. Suponha que $\mathbb{V}$ e $\mathbb{W}$ são dois espaços vetoriais com dimensões m e n respectivamente. Portanto, o produto tensorial $\mathbb{V} \otimes \mathbb{W}$ (lê-se também $\mathbb{V}$ tensor $\mathbb{W}$) produz um espaço com dimensão mn. Vetores em um espaço podem serem reescritos como combinação linear entre os vetores da base desse espaço. Logo, como $\mathbb{V} \otimes \mathbb{W}$ é um espaço vetorial, os vetores nele pode serem escritos como:

$$a_1(|v_1\rangle \otimes |w_1\rangle) + a_2(|v_2\rangle \otimes |w_2\rangle) + ... + a_n(|v_n\rangle \otimes |w_n\rangle) = \sum_i a_i(|v_i\rangle \otimes |w_i\rangle)$$

onde $v_i \in \mathbb{V}$ e $w_i \in \mathbb{W}$.

As seguintes propriedades são satisfeitas por esse operador:

- Multiplicação por escalar ($c \in \mathbb{C}$):
  $c(|v\rangle \otimes |w\rangle) = (c|v\rangle) \otimes |w\rangle = |v\rangle \otimes (c|w\rangle)$

- Adição entre vetores:
  $(|v_1\rangle + |v_2\rangle) \otimes |w\rangle) = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle$
  $|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle$

Uma característica importante do *produto tensorial* é que ele também permite aplicação entre operadores lineares. Seja A um operador linear do espaço vetorial $\mathbb{V}$ e B um operador linear do espaço vetorial $\mathbb{W}$. Logo $A \otimes B$ é um operador linear que age no espaço vetorial $\mathbb{V} \otimes \mathbb{W}$ que é definido como:

$$(A \otimes B)(|v\rangle \otimes |w\rangle) \equiv A|v\rangle \otimes B|w\rangle$$

Um outro fato interessante que pode ser muito útil em determinadas situações é que o *produto tensorial* pode ser definido em termos de produto interno. Assim temos:

$$\mathbb{V} \otimes \mathbb{W} = \left\langle \sum_i a_i|v_i\rangle \otimes |w_i\rangle \middle| \sum_j b_j|w_j\rangle \otimes |w_j\rangle \right\rangle = \sum_{ij} a_i{}^* b_j \langle v_i|v_j\rangle\langle w_i|w_j\rangle$$

Uma representação para produto tensorial é feita através do chamado *Produto de Kronecker*. Esse produto define o produto tensorial entre uma matriz $A_{n\times m}$ e $B_{p\times q}$ como sendo:

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & ... & A_{1n}B \\ A_{21}B & A_{22}B & ... & A_{2n}B \\ \vdots & \vdots & \vdots & \vdots \\ A_{m1}B & A_{m2}B & ... & A_{mn}B \end{bmatrix}_{mp\times nq}$$

que pode ser decomposto da seguinte forma:

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \otimes \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} = \begin{bmatrix} A_1 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \\ A_2 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \\ A_3 \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} A_1B_1 \\ A_2B_2 \\ A_3B_3 \\ A_4B_4 \\ A_5B_5 \\ A_6B_6 \\ A_7B_7 \\ A_8B_8 \\ A_9B_9 \\ A_{10}B_{10} \\ A_{11}B_{11} \\ A_{12}B_{12} \end{bmatrix}$$

A Tabela 1 abaixo contém um resumo dos conceitos até aqui vistos.

Notação	Significado
z^*	Complexo conjugado de um número complexo z.
$\|\psi\rangle$	Vetor (ou estado) conhecido como *ket*.
$\langle\psi\|$	Vetor dual de $\|\psi\rangle$ conhecido como *bra*, que nada mais é que $\|v\rangle^\dagger$.
$\langle\phi\|\psi\rangle$	Produto interno complexo entre os vetores $\|\phi\rangle$ e $\|\psi\rangle$
$\|\phi\rangle \otimes \|\psi\rangle$	Produto tensorial entre os vetores $\|\phi\rangle$ e $\|\psi\rangle$ que podemos abreviar como $\|\phi\psi\rangle$
A^*	Complexo conjugado da matriz A.
A^T	Transposta da matriz A.
$A^\dagger$	Adjunta da matriz A, $A^\dagger = (A^T)^*$

Tabela 1 – Notação de Dirac

4.2 Bits Quânticos

A ambientação quântica impõe que a forma de representa-lo seja diferente da clássica, que é representado por corrente elétrica. Um bit quântico se faz necessário representar através de algum elemento que contenha características quânticas que possam expressar da mesma forma a dualidade de um bit clássico. Um exemplo é utilizar o *spin* de elétrons ou *fótons*.

Com isso é possível trabalhar com a formalização teórica utilizando ferramentas provindas pela física quântica. Segundo um dos postulados da mecânica quântica, qualquer sistema físico tem associado a ele um espaço vetorial complexo onde os possíveis de um sistema são descritos como os vetores existentes nesse espaço vetorial (NIELSEN; CHUANG, 2011). Classicamente os bits podem assumir os valores 0 ou 1, isso também acontece com *qubits* (que são os análogos quânticos do bit), no qual pode assumir os valores $|0\rangle$ ou $|1\rangle$ ou ambos ao mesmo tempo devido ao fenômeno da superposição.

Definição 1. *Um qubit $|\psi\rangle$ é uma combinação linear tal que*

$$|\psi\rangle = a_0|0\rangle + a_1|1\rangle = a_0 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + a_1 \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} a_0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ a_1 \end{bmatrix} = \begin{bmatrix} a_0 \\ a_1 \end{bmatrix}$$

As componentes a_0 e a_1 são números complexos tais que $|a_0|^2 + |a_1|^2 = 1$. O valor $|a_0|^2$ é a probabilidade do *qubit* assumir o estado $|0\rangle$ quando observado, e $|a_1|^2$ a probabilidade do *qubit* assumir o valor $|1\rangle$. Enquanto um *qubit* não é observado, ele existe de forma n-dimensional descrito pelo valor de suas componentes no espaço vetorial correspondente.

É importante frisar que o caso acima é uma definição para um *qubit* de tamanho um. Como há duas bases envolvidas, o espaço vetorial é $\mathbb{C} \times \mathbb{C} = \mathbb{C}^2$. Caso se queira representar um *qubit* de tamanho dois, os possíveis valores de saída e que formariam a base do espaço vetorial seriam dado pelos vetores $|00\rangle, |01\rangle, |10\rangle$ e $|11\rangle$, e o espaço vetorial seria $\mathbb{C}^4$.

Entretanto um computador não é muito útil somente com um ou dois *qubits* para se realizar cálculos. O espaço vetorial envolvido terá sempre dimensão 2^n onde n é o tamanho

do *qubit*. Sendo assim, pode-se generalizar a definição anterior.

Definição 2. *Um qubit de tamanho n pode ser expresso como:*

$$|\psi\rangle^{\otimes n} = \sum_{i=0}^{2^{n-1}} a_i |\beta_i\rangle^{\otimes n} \quad talque \quad \sum_{i=0}^{2^{n-1}} |a_i|^2 = 1.$$

De forma análoga, $|a_i|^2$ representa a probabilidade de o *qubit* quando observado assumir o valor da base $|\beta_i\rangle$ quando medido. Mais detalhes sobre medição serão apresentados na seção 4.2.2.

4.2.1 Operadores quânticos

Diferentemente do caso clássico onde operações sobre bits são feitos através de operações booleanas, aqui como o formalismo é feito utilizando de álgebra linear, uma operação sobre um *qubit* é realizado através da aplicação de alguma matriz unitária, visando alterar a amplitude das probabilidades das componentes. O uso de matriz unitária é feito pois ela sempre possui inversa, tornando assim as transformações sobre *qubits* reversíveis (METODI; CHONG, 2006).

Definição 3. *Um operador U é uma transformação linear que age em um qubit de tamanho n tal que:*

$$U|\psi\rangle^{\otimes n} = U\left(\sum_{i=0}^{2^n-1} a_i |\beta_i\rangle^{\otimes n}\right) = a_i \sum_{i=0}^{2^n-1} U|\beta_i\rangle^{\otimes n} = |\phi\rangle \tag{2}$$

Uma forma interessante e muito útil de construir um operador é escrevendo os elementos em termos de produto interno e, aplicando a operação desejada em cada elemento da matriz. Isso é possível graças a *relação de fechamento* conforme explica McMahon (2007).

Definição 4. *Um operador $U_{n\times n}$ pode ser decomposto em termos de produto interno na forma:*

$$U = \begin{bmatrix} \langle\beta_0|U|\beta_0\rangle & \langle\beta_0|U|\beta_1\rangle & \dots & \langle\beta_0|U|\beta_n\rangle \\ \langle\beta_1|U|\beta_0\rangle & \langle\beta_1|U|\beta_1\rangle & \dots & \langle\beta_1|U|\beta_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle\beta_n|U|\beta_0\rangle & \langle\beta_n|U|\beta_1\rangle & \dots & \langle\beta_n|U|\beta_n\rangle \end{bmatrix}$$

Com a definição acima, suponha que queremos saber a matriz de um operador A se comporta da forma que quando $A|0\rangle = |0\rangle$ e $A|1\rangle = -|1\rangle$. Então podemos escrevê-la da seguinte forma:

$$A = \begin{bmatrix} \langle 0|A|0\rangle & \langle 0|A|1\rangle \\ \langle 1|A|0\rangle & \langle 1|A|1\rangle \end{bmatrix} = \begin{bmatrix} \langle 0|(A|0\rangle) & \langle 0|(A|1\rangle) \\ \langle 1|(A|0\rangle) & \langle 1|(A|1\rangle) \end{bmatrix}$$

$$= \begin{bmatrix} \langle 0|(|0\rangle) & \langle 0|(-|1\rangle) \\ \langle 1|(|0\rangle) & \langle 1|(-|1\rangle) \end{bmatrix} = \begin{bmatrix} \langle 0|0\rangle & -\langle 0|1\rangle \\ \langle 1|0\rangle & -\langle 1|1\rangle \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

4.2.2 Medidas

Um sistema quântico é muito sensível a perturbações externas. Isso faz com que a computação quântica tenha que trabalhar com *qubits* e operações sobre eles sem observar de fato tais mudanças ocorrendo no sistema. Entretanto, em um determinado momento é desejável que se saiba o resultado final de uma operação ou de um algoritmo, sendo assim, torna-se inerente observar o sistema. Entretanto isso possui algumas exigências e implicações. Primeiro é necessário definir o que se quer observar, isto é, qual propriedade associada ao *qubit* deseja-se conhecer. Assim, tal medição é feita por um operador que seleciona a respectiva informação que se pretende obter. Por exemplo, dado um *qubit* $|\psi\rangle = a_0|0\rangle + a_1|1\rangle$, há dois possíveis operadores de medida para ele, que são $M_0 = |0\rangle\langle 0|$ e $M_1 = |1\rangle\langle 1|$ que visam obter as probabilidades do resultado vir a ser $|0\rangle$ ou $|1\rangle$ respectivamente. Assim suas aplicações são feitas da seguinte forma:

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = |a_0|^2$$
$$p(1) = \langle\psi|M_1^\dagger M_1|\psi\rangle = |a_1|^2$$

onde p significa probabilidade. E segundo, uma medição faz com que o *qubit* colapse para um dos estados de sua base, ou seja, deve-se ter ciência que ao observa-lo, embora obtenha-se informações a respeito, faz com que seu estado seja alterado. Seguindo o exemplo anterior, o $|\psi\rangle$ pode colapsar após sua medição para uma de suas bases respeitando a seguinte relação:

$$\frac{M_0|\psi\rangle}{|a_0|} = \frac{a_0}{|a_0|}|0\rangle$$

$$\frac{M_1|\psi\rangle}{|a_1|} = \frac{a_1}{|a_1|}|1\rangle$$

4.3 Circuitos Quânticos

Circuitos quânticos tem por objetivo realizar alguma tarefa da mesma forma que seu análogo clássico. Eles são compostos por um conjunto de operadores que representam as portas-lógicas. Um subconjunto de portas-lógicas dentro de um circuito pode ser chamada de *função*.

4.3.1 Portas-quânticas simples

Existem um conjunto de portas-quânticas que são muito utilizadas na construção de circuitos quânticos. São elas:

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}; \quad X = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}; \quad Y = \begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}; \quad Z = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$$

$$P = \begin{bmatrix}1 & 0\\ 0 & e^{i\theta}\end{bmatrix}; \quad S = \begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix} \quad T = \begin{bmatrix}1 & 0\\ 0 & e^{i\pi/4}\end{bmatrix} \quad R = \begin{bmatrix}cos(\gamma) & -sen(\gamma)\\ sen(\gamma) & cos(\gamma)\end{bmatrix}$$

A porta H é a mais importante para a computação quântica pois ela faz com que o *qubit* seja colocado em superposição dos dois estados, característica chave da computação quântica. Observe:

$$H|0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\ 0\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}\right) = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = |+\rangle$$

$$H|1\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}0\\ 1\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ -1\end{bmatrix} = \frac{1}{\sqrt{2}}\left(\begin{bmatrix}1\\ 0\end{bmatrix} - \begin{bmatrix}0\\ 1\end{bmatrix}\right) = \frac{|0\rangle - |1\rangle}{\sqrt{2}} = |-\rangle$$

$$H|\psi\rangle = a_0 H|0\rangle + a_1 H|1\rangle = a_0\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right) + a_1\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = \left(\frac{a_0 + a_1}{\sqrt{2}}\right)|0\rangle + \left(\frac{a_0 - a_1}{\sqrt{2}}\right)|1\rangle$$

Ambas as matrizes mostradas nessa subseção são unitárias, sendo assim, elas sempre possuem inversas. Consequentemente, a aplicação da matriz inversa de um operador faz com que um *qubit* volte ao estado anterior. Isso acaba sendo uma característica que a computação clássica não possui, a reversibilidade de operações.

4.3.2 Portas-quânticas controladas

Uma classe de portas lógicas quânticas que existem são chamadas *portas-controladas*. Com elas é possível implementar a lógica *if-else* nos circuitos. Sua estrutura padrão consiste em um ou mais *qubits* de controle e um *qubit* alvo, de forma que, quando os *qubits* fixados como controle forem $|1\rangle$ o operador é aplicado ao *qubit* alvo, fazendo com que ele inverta seu valor, caso contrário nada acontece. Entretanto pode se ter diferentes construções para portas controladas onde o principio de ativação pode ser outro e a mudança realizada por ela também seja outra.

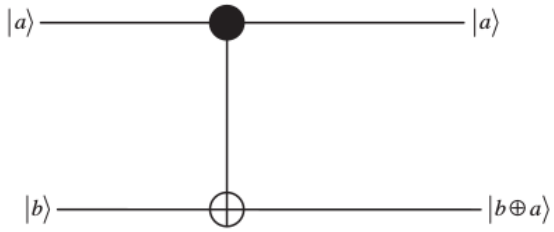

Figura 1 – Porta controlada

A Figura 1 ilustra uma porta-controlada padrão. O círculo preto no barramento superior indica que o *qubit* corrente nele servirá como controle, e o barramento abaixo contendo um círculo em forma de cruz indica que o *qubit* nesse barramento sera o alvo.

Podemos construir o operador ilustrado na Figura 1 seguindo a Definição 4. Seja U a porta-controlada ilustrada. Queremos que nossa porta se comporte da seguinte forma: $U|00\rangle = |00\rangle, U|01\rangle = |01\rangle, U|10\rangle = |11\rangle$ e $U|11\rangle = |10\rangle$. Assim escrevemos U como:

$$U = \begin{bmatrix} \langle 00|U|00\rangle & \langle 00|U|01\rangle & \langle 00|U|10\rangle & \langle 00|U|11\rangle \\ \langle 01|U|00\rangle & \langle 01|U|01\rangle & \langle 01|U|10\rangle & \langle 01|U|11\rangle \\ \langle 10|U|00\rangle & \langle 10|U|01\rangle & \langle 10|U|10\rangle & \langle 10|U|11\rangle \\ \langle 11|U|00\rangle & \langle 11|U|01\rangle & \langle 11|U|10\rangle & \langle 11|U|11\rangle \end{bmatrix}$$

$$= \begin{bmatrix} \langle 00|00\rangle & \langle 00|01\rangle & \langle 00|11\rangle & \langle 00|10\rangle \\ \langle 01|00\rangle & \langle 01|01\rangle & \langle 01|11\rangle & \langle 01|10\rangle \\ \langle 10|00\rangle & \langle 10|01\rangle & \langle 10|11\rangle & \langle 10|10\rangle \\ \langle 11|00\rangle & \langle 11|01\rangle & \langle 11|11\rangle & \langle 11|10\rangle \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad \text{(após aplicação da Função de } \textit{Kronecker}\text{)}$$

A matriz acima também é conhecida como porta *C-NOT*, uma porta muito utilizada na construção de circuitos também.

Uma demonstração interessante da flexibilidade que a arquitetura quântica pode proporcionar é visto com a implementação da porta *Toffoli*. Essa porta é capaz de realizar sozinha 4 operações lógicas clássicas conhecidas, que são: NOT, AND, XOR e NAND. O operador abaixo é o que representa tal porta.

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Sua implementação se dá pela utilização de 3 *qubits* onde os dois primeiros atuam como controle e o último como alvo. Dependendo de que estado se encontra cada um dos *qubits* de entrada, a função implementada pela Toffoli difere. As possíveis configurações são:

- $T|1,1,x\rangle = |1,1,\overline{x}\rangle$ (NOT)
- $T|x,y,0\rangle = |x,y,x \wedge y\rangle$ (AND)
- $T|1,x,y\rangle = |1,x,x \oplus y\rangle$ (XOR)
- $T|x,y,1\rangle = |x,y,\overline{x \wedge y}\rangle$ (NAND)

4.4 Funções

Na computação quântica, funções são computadas de maneira diferente devido o fato de não ser possível realizar um mapeamento um-para-um utilizando somente um *qubit*. A maneira que se adota para contornar esse problema é trabalhar olhando um registrador como se fosse dois distintos, onde o primeiro armazena a entrada e o segundo armazena a saída. Assim, utilizasse um *qubit* auxiliar $|y\rangle = |0\rangle$ para a computação acontecer. Tem-se então:

Definição 5. *Seja f uma função. Define-se o operador U_f como o operador que implementa f que toma como entrada $|x\rangle|y\rangle$ e produz como saída $|x\rangle|y \oplus f(x)\rangle$.*

$$U_f(|x\rangle|y\rangle) = U_f(|x\rangle|0\rangle) = |x\rangle|0 \oplus f(x)\rangle = |x\rangle|f(x)\rangle$$

O símbolo "$\oplus$" significa soma módulo dois, cujo o valor será 1 caso os operandos sejam diferentes e 0 caso sejam iguais. Com a definição de função anterior o problema de mapeamento um-para-um é resolvido pois, se $|x\rangle$ e $|y\rangle$ são levados a $|f(x)\rangle$ a representação deles vetorialmente será diferente, isto é, no primeiro caso será $|x\rangle|f(x)\rangle \neq |y\rangle|f(x)\rangle$. A Figura 2 ilustra a representação gráfica de uma função.

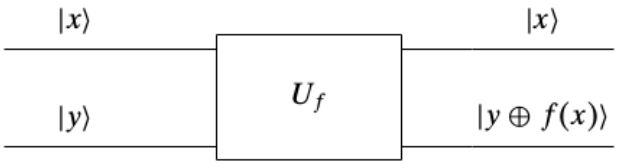

Figura 2 – Exemplo de uma função. Fonte: Yanofsky e Mannucci (2008)

4.5 Complexidade

Na teoria da computação há o estudo da complexidade computacional. Com ela é possível quantificar a quantidade de esforço computacional (recursos) que um algoritmo irá demandar em sua execução. É através dela que é possível de certa forma comparar algoritmos e saber qual é melhor e qual é pior. Classicamente a medida de complexidade computacional se dá pela estimação de quantas instruções que, no pior caso o algoritmo terá que executar. Essa formalização é descrita pela notação $O(f(n))$. Por exemplo, um algoritmo de busca linear, no seu pior caso, isto é, caso o número que deseja encontrar esteja no final do vetor de tamanho n, será necessário percorrer n posições até encontra-lo, portanto, dizemos que esse algoritmo é $O(n)$. Claro que pode haver outros passos no algoritmo, mas a notação admite apenas o valor de maior grau do custo, pois é esse valor que influencia de forma mais intensa o custo final do algoritmo.

Para fins de comparação entre um algoritmo quântico e um análogo clássico essa é a métrica utilizada. Entretanto, entre algoritmos estritamente quânticos outras métricas podem ser utilizadas quantificar a complexidade de um algoritmo que não sejam a quantidade de instruções necessárias no pior caso. (RIEFFEL; POLAK, 2011) comenta algumas métricas de complexidade que podem ser adotadas. São elas:

Quantidade de portas: essa métrica leva em conta a quantidade de portas lógicas existentes no circuito.

Complexidade de consulta: leva em consideração o número de chamadas a uma função para se resolver um determinado problema. Normalmente usado em algoritmos que demandam de iteração.

Complexidade de comunicação: Essa é uma métrica utilizada para algoritmos de transmissão no qual estima a quantidade mínima de *qubits* necessários ser transmitidos para realizar uma tarefa.

4.6 Algoritmos Quânticos

Define-se um algoritmo como um conjunto de instruções para realizar uma determinada tarefa. Um dos objetivos da computação quântica é conseguir projetar algoritmos para tais computadores com um maior nível de eficiência em termos de complexidade

computacional que um análogo em um computador clássico. Para ajudar nessa tarefa, é utilizado a característica da superposição dos estados visando realizar diversos cálculos de uma só vez simultaneamente, isto é, um nível de paralelismo que não é possível realizar em um computador clássico. Yanofsky e Mannucci (2008) definem o um procedimento básico para elaboração de algoritmos quânticos que consiste das seguintes etapas:

1. Tomar como entrada um sistema com n *qubits*.
2. Colocar esses *qubits* em superposição.
3. Aplicar um conjunto de operações unitárias.
4. Medir os *qubits*.

Esse procedimento não é uma regra a ser seguida, apenas uma sugestão pois, assim como na computação tradicional, um algoritmo pode ser implementado de diversas formas para solucionar o mesmo problema. A seguir apresentamos alguns exemplos de algoritmos bem conhecidos no meio quântico.

4.6.1 Algoritmo de Deutsch

O algoritmo de Deutsch tem por objetivo descobrir se uma função $f : \{0,1\} \to \{0,1\}$[1] é balanceada ou constante. Uma função é dita constante se $f(0) = f(1)$, e balanceada se $f(0) \neq f(1)$. Com um computador simples, seria necessário realizar duas operações, calcular $f(0)$ e $f(1)$, entretanto com um computador quântico é possível calcular para ambas as entradas com apenas um cálculo.

Os passos do algoritmo de Deutsch, seguindo o modelo comentado anteriormente, acabam sendo:

1. A entrada do sistema é o *qubit* $|\psi\rangle = |01\rangle$.
2. Colocar ambos os *qubits* em superposição, assim teremos um $|\psi_1\rangle$.

$$H^{\otimes 2}|01\rangle = (H \otimes H)(|0\rangle \otimes |1\rangle) = H|0\rangle \otimes H|1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = |\psi_1\rangle$$

3. Aplica a função f em $|\psi_1\rangle$, assim, tem-se um $|\psi_2\rangle$.

$$U_f|\psi_1\rangle = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0 \oplus f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}\right) = \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|f(x)\rangle - |\overline{f(x)}\rangle}{\sqrt{2}}\right)$$

[1] A simbologia $\{0,1\}^n$ vem da área de Linguagens Formais. Neste caso simboliza o conjunto de todas as palavras de tamanho n formada pela combinação de 0's e 1's. A função f definida significa que ela toma como entrada uma palavra (*qubit*) de tamanho um e produz como saída também uma palavra de tamanho um (assumimos n como 1 quando não é explícito). Ideia análoga a domínio e contra-domínio.

$$= \begin{cases} \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right), & \text{se } f(x) = 0 \\ \left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|1\rangle - |0\rangle}{\sqrt{2}}\right), & \text{se } f(x) = 1 \end{cases}$$

OBS: $\overline{f(x)}$ significa o oposto de $f(x)$. Sabendo que $a - b = (-1)(b - a)$ podemos simplificar o resultado anterior, utilizando da propriedade de multiplicação por escalar no produto tensorial, de forma que:

$$= (-1)^{f(x)}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}}\right)\left(\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right) = (-1)^{f(x)}|+\rangle|-\rangle = |\psi_2\rangle$$

4. Aplica H no primeiro *qubit*, resultando em um $|\psi_3\rangle$. Assim, a simplificação feita no item anterior ilustra que o primeiro *qubit* é afetado pelo possível resultado do segundo. Assim tem-se:

$$(H \otimes I)|\psi_2\rangle = \begin{cases} (\pm 1)(H|+\rangle) \otimes (I|-\rangle), & \text{se } f \text{ é constante;} \\ (\pm 1)(H|-\rangle) \otimes (I|+\rangle), & \text{se } f \text{ é balanceada.} \end{cases}$$

concluindo que:

$$|\psi_3\rangle = \begin{cases} (\pm 1)|0\rangle|-\rangle, & \text{se } f \text{ é constante;} \\ (\pm 1)|1\rangle|+\rangle, & \text{se } f \text{ é balanceada.} \end{cases}$$

Embora pareça que $|x\rangle$ permaneça inalterado do início ao fim, sua iteração com o sistema como um todo é crucial para o funcionamento do algoritmo. Assim, observando somente o primeiro *qubit* no final do algoritmo é possível saber se a função é constante ou balanceada.

Resumindo os procedimentos de forma matricial, tem-se $(H \otimes I)U_f(H \otimes H)|01\rangle$ e, a Figura 3 ilustra o algoritmo em termo de circuito quântico.

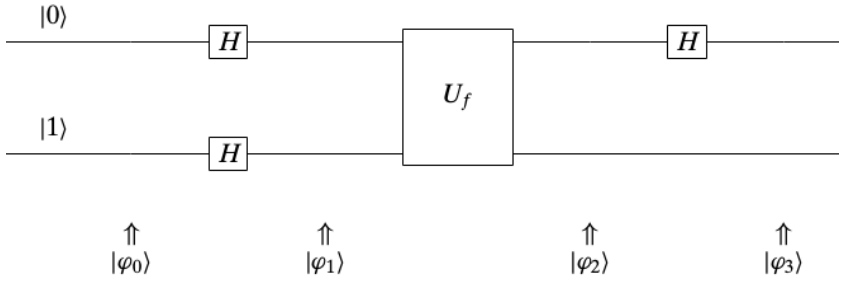

Figura 3 – Circuito que implementa o algoritmo de Deutsch. Fonte: Yanofsky e Mannucci (2008)

Esse algoritmo pode ser considerado um caso particular do algoritmo de *Deutsch-Jozsa* onde, tal algoritmo funciona considerando uma função $f : \{0,1\}^n \rightarrow \{0,1\}$. O modo de funcionamento é análogo ao algoritmo de *Deutsch* mostrado, porém considerando uma entrada de tamanho n onde os $n-1$ *qubits* serão $|0\rangle$ e somente o último é $|1\rangle$, isto é, a entrada será $|0\rangle^{\otimes n-1}|1\rangle$. Consequentemente, a aplicações se mantém as mesmas. Ao final é medido os $n-1$ *qubits* e, se a função for constante o resultado será $|0\rangle^{\otimes n-1}$, caso a saída seja diferente disso, a função é considerada balanceada.

4.6.2 Algoritmo de Grover

O algoritmo de Grover é um algoritmo para realizar busca em um conjunto de dados. Ele é mais elaborado que os exemplos anteriores porquê ele necessita fazer iterações, ou seja, uma sequência de passos será repetida até que uma condição de parada seja satisfeita. Diferentemente de alguns algoritmos de busca clássicos onde é utilizado algum tipo de sequencialidade como o *selection-sort* ou *insertion-sort*, ou uma estratégia de divisão e conquista como *merge-sort* ou *quick-sort*, o algoritmo de Grover possui um caráter diferenciado, ele funciona de forma desordenada. A Figura 4 ilustra a representação em forma de circuito de tal algoritmo. Abaixo segue os passos de funcionamento.

1. A entrada do algoritmo é $|0\rangle^{\otimes n}|1\rangle$.
2. Coloca-se todos os bits da entrada em superposição com aplicação de uma $H^{\otimes n}$.

$$H^{\otimes n}|0\rangle^{\otimes n}|1\rangle = H|0\rangle \otimes H|0\rangle \otimes H|0\rangle \otimes \ldots \otimes H|1\rangle = |+\rangle^{\otimes n}|-\rangle$$

3. Repetir $\sqrt{2^n}$ vezes.
   - Aplicar uma inversão de fase $U_f(I^{\otimes n} \otimes H)$. Com o operador U_f seleciona-se apenas a posição correspondente de valor igual a palavra que se deseja capturar. Com a inversão de fase faz com que apenas tal posição tenha o sinal da amplitude invertido.
   - Aplicar o operador $-I + 2A$. Esse é um truque utilizado no algoritmo que faz com que a amplitude das posições que estão com a fase invertida seja ampliada e invertida novamente, e as que estão com a fase positiva tenham a amplitude diminuída. Em termos geométricos isso faz com que o vetor vá se aproximando cada vez mais da base que se deseja selecionar (que é correspondente a palavra de busca definida na construção do operador U_f.
4. Após isso, mede-se os *qubits* de entrada, e como resultado, ele irá colapsar com uma alta probabilidade para a base correspondente a palavra.

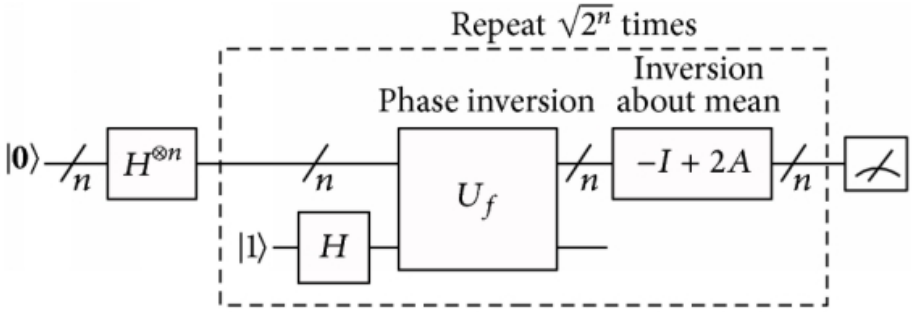

Figura 4 – Circuito que implementa o algoritmo de Grover. Fonte: Yanofsky e Mannucci (2008)

5 DISCUSSÃO

Pelo fato da inerente necessidade de mudar o paradigma de computação para um novo que suporte a demanda de processamento computacional, espera-se que nesse novo paradigma um computador seja capaz de prover este ganho. A computação quântica ao que tudo indica tem essa capacidade. Algoritmos construídos para um computador quântico se mostram mais eficientes que os realizados por computação clássica. Por exemplo, o algoritmo de *Deutsch–Jozsa* realiza sua tarefa em tempo $O(1)$, enquanto um análogo clássico levaria um tempo $O(2^n)$, um ganho absurdamente grande pois é difícil achar otimizações que façam a complexidade reduzir de exponencial para constante. Já o algoritmo de Grover demanda um tempo $O(\sqrt{2^n})$ enquanto um análogo clássico tome tempo $O(2^n)$. Esse resultado proporcionado com o algoritmo de Grover é provado matematicamente que é o melhor que se pode obter com uma busca desordenada, isto é, qualquer outra possível implementação será igual ou pior que ela, conforme explica Rieffel e Polak (2011). Outro exemplo interessante que vale a pena comentar embora porém que não foi mostrado neste trabalho – devido sua exigência de conceitos que fogem do escopo deste trabalho, é o algoritmo de Shor. O algoritmo de Shor é um algoritmo de fatoração de números primos. Esse algoritmo consegue fatorar números primes muito grandes em tempo polinomial, enquanto classicamente essa tarefa demanda tempo exponencial.

Espera-se que, conforme a evolução ocorra, que computadores quânticos sejam capaz de auxiliar a ciência realizando simulações de sistemas quânticos pois, atualmente, tais simulações feitas em computadores clássicos demandam muito recurso computacional. Entretanto há uma preocupação também entre a comunidade científica quanto a ética do uso desse tipo de máquina. Isto ocorre devido o fato de os principais sistemas de criptografia existentes atualmente utilizarem do princípio de fatoração de números primos muito grandes, e Shor mostrou que, com um computador quântico, esses fatores podem ser encontrados rapidamente, fazendo com que a maioria dos sistemas criptografados corram perigo de serem quebrados.

Com a questão de segurança em cheque, já há uma preocupação em desenvolver técnicas de criptografia quântica. Tais técnicas tentam utilizar da característica sensível que um sistema quântico possui, isto é, que um sistema quântico não pode ser observado sem que tenha seu estado alterado (NIELSEN; CHUANG, 2011).

Percebe-se que, a evolução dos sistemas computacionais para um paradigma quântico envolvem não somente questões relacionadas a eficiência, mas também a questões relacionadas a uma fragilidade que será gerada que impacta diretamente a computação clássica, como ética e segurança, questões essas que não podem ser deixadas em segundo plano com o desenvolvimento dessa nova tecnologia.

6 CONCLUSÕES

Este projeto teve como objetivo abordar questões centrais relacionadas a computação e informação quântica com uma abordagem mais simplificada, de forma que os leitores que nunca tiveram contato com tais temas sejam introduzidos a eles de forma simples, direta e objetiva.

Embora haja muitas semelhanças entre computação quântica e computação clássica, há diversos outros conceitos que devem ser estudados para se trabalhar com computação quântica, principalmente uma solidez em conceitos de álgebra linear pois, é com ela que toda a abstração da computação é feita. Com o conhecimento em tais conceitos torna possível saber como utilizar das características que o meio quântico possui a fim de se projetar algoritmos que sejam mais eficientes que os clássicos, para que assim, se justifique o desenvolvimento dessa tecnologia, a fim de prover meios ao ser humano para que novas descobertas sejam feitas através de sua capacidade computacional, fazendo assim que se traga melhorias a sociedade.

Referências

MCMAHON, D. **Quantum Computing Explained**. New Jersey: Hoboken, 2007.

METODI, T. S.; CHONG, F. T. **Quantum Computing for Computer Architects (Synthesis Lectures on Computer Architecture)**. San Rafael: Morgan and Claypool Publishers, 2006.

NIELSEN, M. A.; CHUANG, I. L. **Quantum Computation and Quantum Information: 10th Anniversary Edition**. New York: Cambridge University Press, 2011.

RIEFFEL, E. G.; POLAK, W. H. **Quantum Computing: A Gentle Introduction (Scientific and Engineering Computation)**. Massachusetts: The MIT Press, 2011.

YANOFSKY, N. S.; MANNUCCI, M. A. **Quantum Computing for Computer Scientists**. New York: Cambridge University Press, 2008.